from the execution of the copy of the software by the corresponding platform, can be received from each of the multiple platforms. At the operation **1304**, the number of matching results of execution of the software can be determined. At the operation **1306**, whether the number of the matching results is equal to or greater than the first threshold can be determined. At the operation **1308**, the matching results can be accepted as the agreed-upon output of the software in response to the number of the matching results being equal to or greater than the first threshold. For example, an entity may desire to use a particular software that the entity knows has a specific hash. The entity can send requests to multiple platforms to determine whether each platform has software available for execution by the platform that has the same hash. The entity can then select the platforms to use for execution of the software from among those that provide a response indicating that the software having the desired hash is available on that platform.

FIG. **17** is a flow diagram illustrating an example of a method **1700** for causing software to be executed. In the method **1700**, at an operation **1702**, a message can be sent to each of multiple platforms. The message can include an input for the software. At the operation **1302**, the result, produced from the execution of the copy of the software by the corresponding platform, can be received from each of the multiple platforms. At the operation **1304**, the number of matching results of execution of the software can be determined. At the operation **1306**, whether the number of the matching results is equal to or greater than the first threshold can be determined. At the operation **1308**, the matching results can be accepted as the agreed-upon output of the software in response to the number of the matching results being equal to or greater than the first threshold.

FIG. **18** is a flow diagram illustrating an example of a method **1800** for causing software to be executed. In the method **1800**, at an operation **1802**, at least one copy of the software can be sent to at least one of multiple platforms. At the operation **1302**, the result, produced from the execution of the copy of the software by the corresponding platform, can be received from each of the multiple platforms. At the operation **1304**, the number of matching results of execution of the software can be determined. At the operation **1306**, whether the number of the matching results is equal to or greater than the first threshold can be determined. At the operation **1308**, the matching results can be accepted as the agreed-upon output of the software in response to the number of the matching results being equal to or greater than the first threshold.

FIG. **19** is a flow diagram illustrating an example of a method **1900** for causing software to be executed. In the method **1900**, at an operation **1902**, multiple platforms can be selected from a population of platforms. At the operation **1302**, the result, produced from the execution of the copy of the software by the corresponding platform, can be received from each of the multiple platforms. At the operation **1304**, the number of matching results of execution of the software can be determined. At the operation **1306**, whether the number of the matching results is equal to or greater than the first threshold can be determined. At the operation **1308**, the matching results can be accepted as the agreed-upon output of the software in response to the number of the matching results being equal to or greater than the first threshold.

In an aspect, the software can be caused to be executed using any combination of the operations of the methods **1300**, **1400**, **1500**, **1600**, **1700**, **1800**, and **1900**.

More generally, various aspects of the presently disclosed subject matter can include or be embodied in the form of computer-implemented processes and apparatuses for practicing those processes. Aspects also can be implemented in the form of a computer program product having computer program code containing instructions embodied in non-transitory and/or tangible media, such as floppy diskettes, compact disc read-only memory (CD-ROMs), hard drives, universal serial bus (USB) drives, or any other machine readable storage medium, such that when the computer program code is loaded into and executed by a computer, the computer becomes an apparatus for practicing aspects of the disclosed subject matter. Aspects also can be implemented in the form of computer program code, for example, whether stored in a storage medium, loaded into and/or executed by a computer, or transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, or via electromagnetic radiation, such that when the computer program code is loaded into and executed by a computer, the computer becomes an apparatus for practicing aspects of the disclosed subject matter. When implemented on a general-purpose microprocessor, the computer program code segments can configure the microprocessor to produce specific logic circuits.

In some configurations, a set of computer-readable instructions stored on a computer-readable storage medium can be implemented by a general-purpose processor, which can transform the general-purpose processor or a device containing the general-purpose processor into a special-purpose device configured to implement or carry out the instructions. Aspects can be implemented using hardware that can include a processor, such as a general purpose microprocessor and/or an application-specific integrated circuit (ASIC) that implements all or part of the techniques according to aspects of the disclosed subject matter in hardware and/or firmware. The processor can be coupled to memory, such as random-access memory (RAM), read-only memory (ROM), flash memory, a hard disk or any other device configured to store electronic information. The memory can store instructions adapted to be executed by the processor to perform the techniques according to aspects of the disclosed subject matter.

The foregoing description, for purpose of explanation, has been described with reference to specific aspects. However, the illustrative discussions above are not intended to be exhaustive or to limit aspects of the disclosed subject matter to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The aspects were chosen and described in order to explain the principles of aspects of the disclosed subject matter and their practical applications, to thereby enable others skilled in the art to utilize those aspects as well as various aspects with various modifications as may be suited to the particular use contemplated.

The invention claimed is:

1. A method for verifying a sequence of instructions of a software, comprising:

receiving, at a first electronic device from a first computer platform, a first hash of a first copy of the software, the first copy of the software having been received by the first computer platform from a second electronic device, the first computer platform configured to execute the first copy of the software;

receiving, at the first electronic device from the second electronic device, a second copy of the software;

producing, at the first electronic device, a second hash of the second copy of the software;

comparing, at the first electronic device, the first hash and the second hash;